

---

# Implicit Surface Modelling as an Eigenvalue Problem

---

Christian Walder  
Olivier Chapelle  
Bernhard Schölkopf

Max Planck Institute for Biological Cybernetics  
Spemannstr. 38, 72076 Tuebingen, Germany

CHRISTIAN.WALDER@TUEBINGEN.MPG.DE  
OLIVIER.CHAPELLE@TUEBINGEN.MPG.DE  
BERHARD.SCHOELKOPF@TUEBINGEN.MPG.DE

## Abstract

We discuss the problem of fitting an implicit shape model to a set of points sampled from a co-dimension one manifold of arbitrary topology. The method solves a non-convex optimisation problem in the embedding function that defines the implicit by way of its zero level set. By assuming that the solution is a mixture of radial basis functions of varying widths we attain the globally optimal solution by way of an equivalent eigenvalue problem, without using or constructing as an intermediate step the normal vectors of the manifold at each data point. We demonstrate the system on two and three dimensional data, with examples of missing data interpolation and set operations on the resultant shapes.

## 1. Introduction

The problem of inferring a co-dimension one<sup>1</sup> manifold  $\mathcal{X} \subset \mathbb{R}^d$  from a finite sampling  $\mathcal{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathcal{X}\}$  frequently arises in the domain of computer graphics and machine vision. Most commonly one deals with three dimensional data from a laser scanner or optical triangulation device, however the two dimensional case of inferring contours and the four dimensional case of estimating a three dimensional shape evolving over time are also common.

Although piece-wise linear approximations of  $\mathcal{X}$  (*i.e.*

---

<sup>1</sup>That is a manifold of dimensionality one less than that of the space in which it lies.

---

Appearing in *Proceedings of the 22<sup>nd</sup> International Conference on Machine Learning*, Bonn, Germany, 2005. Copyright 2005 by the author(s)/owner(s).

triangulated meshes) are the most common representation, recent years have seen increasing interest in the use of implicit shape models (Carr et al., 2001; Walder et al., 2003; Shen et al., 2004; Schölkopf et al., 2005). Implicit shape models (or simply *implicit*s) use an embedding function  $f : \mathcal{X} \rightarrow \mathbb{R}$ , that defines the hyper-surface implicitly by way of its zero level set  $f^{-1}(0)$ . Such models can smoothly interpolate holes in the data, and a number of derivatives of  $f$  will typically exist, which can be used for analysis (see for example (Ohtake et al., 2004)). Additionally, as the sign of  $f$  indicates whether a given point is interior or exterior to the surface (see Figure 1), collision detection and other set operations between shapes can be implemented easily and efficiently.

Most of the work on implicit as we have defined them can be divided into two classes. The first class does not separate inferring  $f$  from (or in other words fitting  $f$  to) the data and evaluating  $f$  – the complete data-set is retained and analysed locally each time  $f$  is evaluated. This approach is simple and effective (Alexa et al., 2001; Shen et al., 2004), but the local nature of the analysis necessitates the use of normal vectors of the target surface  $\mathcal{X}$  at each sample point in order to correctly orientate the implicit.

In the other category, a function  $f$  is inferred from the data, hopefully serving as a compact representation of the key geometrical information without requiring the data itself for evaluation. Typically these methods use a mixture of radial basis functions, either fully supported as in (Carr et al., 2001) or compactly supported as in (Ohtake et al., 2003). Although it is not strictly necessary, all of the methods we are aware of in this class other than that of Schölkopf et al. (2005) also require the surface normal vector at each data point. The present work also falls into this category, and also without using normal vectors.

The decision to use normal vectors is a pragmatic one – it tends to make the fitting process simpler since the normals contain a great deal of information about the target surface, even if this information is somewhat redundant when one has the data points. In some cases however it is not possible to derive these normal vectors reliably. In the case of sparse or high dimensional data for example, deriving the normal vectors (by some local processing) as an intermediate step to the final fitting process could introduce more difficulties than it avoids. To quote the documentation of the state of the art commercially available software product based on the work of Carr et al. (2001): “in general, without additional information determining normals [from points] is ambiguous”.

It turns out that there are few implicit shape fitting algorithms that do not use normal vectors, and the present work is inspired by one of these few, namely the Slab SVM (Schölkopf et al., 2005). The reason that few authors have taken this approach is probably due in part to the fact that, as we shall see, most natural formulations of the problem will be non-convex. To deal with this our method uses only quadratic terms in the optimisation problem in order to derive an equivalent eigenvalue problem that can be solved both optimally and efficiently.

The paper is structured as follows: in the next Section we review the Slab SVM, particularly identifying the improvements that we are aiming for, before describing in detail our new algorithm in the following Section 3. We then demonstrate the algorithm on both two and three dimensional problems in Section 4, before finishing with some concluding remarks in Section 5.

## 2. Related Work

Another method that does not use normal vectors is the Slab SVM – a generalisation of the one-class SVM that essentially applies the “maximum margin” regulariser to the problem of implicit surface modelling (Schölkopf et al., 2005). This is a natural application of kernel methods, and a related approach was taken in (Walder et al., 2003) which generalises the SVM classifier – an alternative which we briefly introduce after the following the description of the Slab SVM.

For the sake of clarity we shall take a simplified view of the Slab SVM – we replace the  $\epsilon$ -insensitive loss with a simple one-norm loss, and assume that a radial basis function (RBF) kernel is being used. In this case the algorithm is similar to a one-class SVM, however the data incur a penalty proportional to  $\sum_i |f(\mathbf{x}_i)|$ , rather than the usual term  $\sum \max(0, f(\mathbf{x}_i))$ . It turns out that

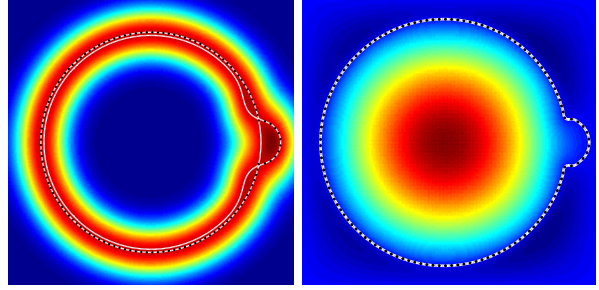


Figure 1. The Slab SVM embedding function (*left*) is reasonable for 3D rendering, however tends to a negative value both inside and outside the surface of interest, whereas the present method (*right*) indicates by way of the sign of the resultant embedding function whether a given point is interior to the surface. The dots are the 2D input data points, the colours represent function value, and the lines depict the zero level set.

the optimisation problem it solves is the following:

$$\arg \min_{f \in \mathcal{H}} \|f\|_{\mathcal{H}}^2 + C \sum_{i=1}^m |f(\mathbf{x}_i)| - \text{const}(f)$$

where  $\text{const}(f)$  denotes the zero order component of the function  $f$  (usually denoted  $b$  or  $\rho$  in the SVM literature), and  $\mathcal{H}$  is a reproducing kernel Hilbert space (RKHS) such that the term  $\|f\|_{\mathcal{H}}^2$  in the above formulation acts as a regulariser. From this perspective we can view this as a variant of a regression problem that has target values of zero at each of the  $\mathbf{x}_i$ . Clearly such a regression problem is solved trivially by the function  $f = 0$ , but the Slab SVM avoids this by including the term  $\text{const}(f)$  in the objective function.

The above approach is advantageous insofar as it yields a convex optimisation problem, however the maximisation of  $\text{const}(f)$  can cause some problems – as a result of it, rather than being positive inside and negative outside the manifold of interest, if an RBF kernel is used this property will often only hold within some local neighbourhood of the data points, as in Figure 1. This causes no problems in rendering an implicit in three dimensions (since the extraneous zero set will always be obscured), but the property that the sign of the function indicates whether a given point is inside the shape no longer holds. The method of Walder et al. (2003), on the other hand, is similar to the above but rather than including the term  $\text{const}(f)$  in the objective, introduces inequality constraints that force the function to be greater/less than the value plus/minus one at some additional points interior/exterior to the target manifold – thereby typically requiring normal vectors in order to derive these additional points.

The goal of the present work then, is to preserve the strengths of the Slab SVM (especially the fact that

it does not use normal vectors) while producing embedding functions with the behaviour depicted on the right side of Figure 1, as these are more suitable for inside/outside tests *etc.* Also, by choosing a regularisation rather than an RKHS perspective, we give an improved treatment of multi-scale basis functions.

### 3. Algorithm

The method we propose can also be viewed as a variant of a regression problem in which all the target values are zero. Essentially we aim for a means of avoiding the triviality of minimising  $\sum_i f(\mathbf{x}_i)^2$  with respect to  $f$  that produces embedding functions whose sign behaves in the manner described previously. The approach we propose is based on the following observations, that are generally true of such embedding functions:

1. The values  $|f(\mathbf{x}_i)|$  are relatively small (data term).
2. The value  $|f(\mathbf{x})|$  is relatively large over most of the space  $\mathcal{X}$  (energy term).
3. The gradients  $\|(\nabla f)(\mathbf{x}_i)\|$  are relatively large (gradient term).
4.  $f$  is smooth (regularisation term).

With the exception of the third point, the above objectives can be likened to a number of methods proposed for semi-supervised learning. In this analogy, we relate the first item to the classification function targets on the labelled points, and the second item to the treatment of the unlabelled points. We mention this to illustrate the non-trivial nature of the problem – similar difficulties arise as in the semi-supervised case. In particular the second item tends to lead to difficult non-convex optimisation problems as in the transductive Support Vector Machine (Vapnik, 1998) for example.

To deal with this we can either set up a difficult non-convex optimisation problem and then apply heuristics to solve it, or instead opt for an easier to solve (although less ideal) problem that can be solved optimally. Here we take the latter approach, in particular we devise four quadratic penalties motivated by the four objectives above, thereby leading to an eigenvalue problem. To this end, we propose to minimise with respect to  $f \in \mathcal{F}$  (to be defined shortly) the functional

$$\begin{aligned} \Theta(f) = & \underbrace{\sum_{i=1}^m f(\mathbf{x}_i)^2}_{\text{data term}} + \underbrace{\lambda_{\Omega} < f, f >_{\mathcal{F}}^2}_{\text{regularisation term}} - \\ & \underbrace{\lambda_e \int_{\mathbf{u} \in \mathbb{R}^d} f(\mathbf{u})^2 d\mu(\mathbf{u})}_{\text{energy term}} - \underbrace{\lambda_{\nabla} \sum_{i=1}^m \|(\nabla f)(\mathbf{x}_i)\|^2}_{\text{gradient term}} \end{aligned} \quad (1)$$

Here we have noted the correspondence between the four terms of the objective function (balanced by the positive real-valued  $\lambda$ 's) and the properties we listed at the start of the section. The terms themselves each appear in quadratic form in order to allow us to formulate the optimisation as an eigenvalue problem. To do this, we must also define the function class as a mixture of radial basis functions – that is, we let  $f \in \mathcal{F}$  admit the form:

$$f(\mathbf{x}) = \sum_{j=1}^n \pi_j \phi(\|\mathbf{v}_j - \mathbf{x}\|/\sigma_j^2) \quad (2)$$

for some set of basis centres  $\{\mathbf{v}_j\}_{1 \leq j \leq n}$  with associated widths  $\sigma_j$ , and basis function  $\phi : \mathbb{R}^+ \rightarrow \mathbb{R}$ . This type of approximation is widely known as the finite element method of approximately solving integro-differential equations, frequently used to calculate heat flows, electro-magnetic fields and material stresses. We shall return to the choice of  $\phi$ , presently we rewrite the objective function in  $\pi$ . Letting  $[K_{xv}]_{i,j} = \phi(\|\mathbf{v}_j - \mathbf{x}_i\|/\sigma_j^2)$ , the data term is:

$$\sum_{i=1}^m f(\mathbf{x}_i)^2 = \pi^T K_{vx}^T K_{vx} \pi$$

Next, due to the quadratic nature of the energy term as well as the linearity of integration, we can write the energy term as a vector-matrix-vector product in the following way. Indeed, let us define  $K_e$  as

$$[K_e]_{i,j} = \int_{\mathbf{u} \in \mathbb{R}^d} \phi(\|\mathbf{v}_j - \mathbf{u}\|/\sigma_j^2) \phi(\|\mathbf{v}_i - \mathbf{u}\|/\sigma_i^2) d\mu(\mathbf{u}).$$

Then,

$$\begin{aligned} \int_{\mathbf{u} \in \mathbb{R}^d} f(\mathbf{u})^2 d\mu(\mathbf{u}) &= \int_{\mathbf{u} \in \mathbb{R}^d} \left( \sum_j \pi_j \phi(\|\mathbf{v}_j - \mathbf{u}\|/\sigma_j^2) \right)^2 d\mu(\mathbf{u}) \\ &= \sum_{j,k} \pi_j \pi_k [K_e]_{j,k} \\ &= \pi^T K_e \pi \end{aligned}$$

Similarly we can rearrange the gradient term to get  $\sum_{i=1}^m \|\nabla_{\mathbf{x}=\mathbf{x}_i} f(\mathbf{x})\|^2 = \pi^T K_{\nabla} \pi$ , defining  $[K_{\nabla}]_{i,j}$  as

$$\sum_{i=1}^m \langle \nabla_{\mathbf{x}=\mathbf{x}_i} \phi(\|\mathbf{x} - \mathbf{v}_j\|/\sigma_j^2), \nabla_{\mathbf{x}=\mathbf{x}_i} \phi(\|\mathbf{x} - \mathbf{v}_k\|/\sigma_k^2) \rangle$$

The only remaining term is the regulariser  $\|f\|_{\mathcal{F}}^2$ , which we discuss in the following sub-section.

### 3.1. Regulariser

Here we discuss and motivate our choice of regulariser. In the first part we indicate some difficulties in using an RKHS norm as a regulariser when one wishes to use basis functions of various widths. We then outline a principled approach to achieving this before proposing a simpler alternative.

#### 3.1.1. REGULARISATION IN AN RKHS

The regulariser that arises in the SVM has already been used to good effect in implicit shape modelling (Walder et al., 2003; Schölkopf et al., 2005), so it seems natural to apply it here. Assuming we have a constant basis function width so that  $\sigma_i = \gamma$ , it is straightforward to define the SVM regulariser  $\Omega(f)$  corresponding to the RKHS<sup>2</sup> of the reproducing kernel  $k_\gamma(\mathbf{x}_i, \mathbf{x}_j) \doteq \phi(\|\mathbf{x}_i - \mathbf{x}_j\|/\gamma^2)$ :

$$\Omega(f) = \sum_{i,j=1}^n \pi_i \pi_j \phi(\|\mathbf{v}_i - \mathbf{v}_j\|/\gamma^2) \quad (3)$$

But if the  $\sigma_j$  are not constant we cannot write down an analogous regulariser the same way. The approach taken in the Slab SVM is to fit an embedding function on a course scale, and then to calculate another embedding function (to be added to the first) using the residual values, rather than 0, as the regression target. This is somewhat suboptimal however, since the individual regularisation terms added together will typically only upper bound the regularisation of the final embedding function. Assuming that the regulariser is a norm of the function, we can see this from the triangle inequality  $\|f + g\|^2 \leq \|f\|^2 + \|g\|^2$ . To put it in intuitive terms, if the course scale solution is concave in a given region, an incentive should be given at finer scales to be convex to the same degree over the same region, whereas the method in question actively penalises this behaviour.

#### 3.1.2. FROM KERNEL TO REGULARIZATION OPERATOR

More accurate is to choose a given scale of basis function  $\gamma$ , at which to induce a regulariser via Equation 3. It turns out (see for example the work of Girosi et al. (1993) and Schölkopf and Smola (2002)) that we can write  $\Omega(f) = \langle \Upsilon f, \Upsilon f \rangle$  for some regularisation operator  $\Upsilon$ . In the present framework, we use the same type of rearrangement as with the energy and derivative terms, which allows us to write  $\Omega(f) = \pi^\top K_\Omega \pi$ ,

<sup>2</sup>Note that for such an RKHS with reproducing kernel  $k_\gamma(\cdot, \cdot)$  to exist it must satisfy the conditions given by Bochner's theorem. In particular, it is sufficient that  $\phi(\cdot)$  has a non-negative Fourier transform.

where  $K_\Omega$  is defined element wise by:

$$[K_\Omega]_{i,j} = \langle \Upsilon f_i, \Upsilon f_j \rangle$$

and we have defined  $f_k(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{v}_k\|/\sigma_k^2)$ . An analysis in (Schölkopf & Smola, 2002) of the regularisation operator of a translation invariant reproducing kernel function shows an interesting interpretation in the Fourier domain – in signal processing terms the regulariser computes the energy of the function after attenuation by some filter function (determined by the choice of kernel function) in the frequency domain. Following this exposition, it turns out that we can write

$$\langle \Upsilon f_i, \Upsilon f_j \rangle_{\mathcal{D}} = \int \frac{\overline{F[f_i](\omega)} F[f_j](\omega)}{F[f_\gamma](\omega)} d\omega \quad (4)$$

where  $F$  is the (forward) Fourier transform operator,  $\bar{x}$  is the complex conjugate of  $x$ , and we have defined  $f_\gamma(\mathbf{x}) = \phi(\|\mathbf{x}\|/\gamma^2)$ . As an aside, it is interesting to consider applying this use of multi scale basis functions in combination with the described regularisation method, to the SVM classifier problem, in order to trade between optimality and computation time in large scale problems – one could imagine using finer width basis functions nearer to the decision boundary. In fact, a related exposition in which basis function and regulariser are separated has lead to a generalised framework that includes a number of learning algorithms as special cases (Smola & Schölkopf, 1997).

#### 3.1.3. SEPARATION OF REGULARISER AND BASIS

Unfortunately however, while some cases can be analytically solved the integral of Equation 4 is, roughly speaking, arbitrarily difficult to solve for an arbitrary kernel function. Thus, we adopt a simpler regularisation operator in order to provide a cleaner treatment of the multi-scale case. The one we choose, already applied to good effect in the context of implicit surfaces (Carr et al., 2001), and analysed some time ago as a regulariser (or stabiliser) in the context of approximation theory by Duchon (1976) is the following *thin plate* energy:

$$\Omega(f) = \int \|\nabla \nabla^\top f(\mathbf{x})\|_F^2 d\mu(\mathbf{x})$$

( $\|\cdot\|_F$  denotes the Frobenius matrix norm), which derives its name from the fact that in two dimensions it is equal to:

$$\int \left( \left( \frac{\partial^2 f}{\partial x^2} \right)^2 + \left( \frac{\partial^2 f}{\partial y^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2 \right) dx dy$$

which approximates the bending energy of the thin metal plate with height  $f(x, y)$ . Using the same rearrangement as before it is easy to write this as a

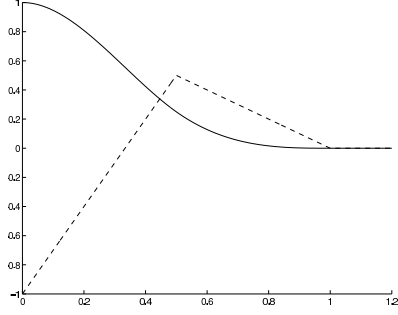


Figure 2. The  $B_3$  spline (solid line) and its second derivative scaled by a factor one on twelve:  $\frac{\partial^2 B_3(r)}{12\partial r^2}$  (dashed line). It is the integral of the squared second derivative that we use as a regulariser. The support of the function is  $[-1, 1]$ .

quadratic term in  $\pi$ , and the resultant matrix has the same sparsity structure as  $K_e$ . Note that if there is a small enough number of different values for the  $\sigma_j$  (as in all our experiments), we can efficiently compute  $K_e$  and  $K_\Omega$  by precomputing the integral for a range of values of  $\|\mathbf{v}_j - \mathbf{v}_k\|$  and for each unique value of  $\sigma_j$ , and then use the results to construct the required matrices. Numerically integrating in this manner introduces no computational disadvantage as the integrals can each be done once and stored for later reuse.

### 3.2. Basis Function

The use of compactly supported basis functions  $\phi(\cdot)$  is attractive as it leads to sparse matrices – reducing computational and storage requirements. Furthermore, as, unlike most authors, we have separated the choice of basis function from the choice of regulariser, our choice of basis constitutes nothing more than a means of trading between efficiency and accuracy. As such, the basis function need only satisfy some quite basic requirements such as not lying in the null space of our regulariser. The basis function we choose then is the so-called  $B_d$ -spline function, which is defined recursively by  $B_0(r) = u(r)$ , where  $u(\cdot)$  is the centred heaviside function, and  $B_d(r) = \bigotimes_{i=1}^{d+1} B_0(r)$ , where  $\bigotimes$  denotes convolution. Interestingly one can show that the  $B_d$  spline approaches the Gaussian as  $d \rightarrow \infty$ . For our experiments, we used the  $B_3$  spline (see Figure 2), which can be evaluated using the equation:

$$B_d(r) \propto \sum_{n=0}^{d+1} \frac{(-1)^n}{d!} \binom{n}{d+1} \left(r + \left(\frac{d+1}{2} - n\right)\right)_+^d$$

### 3.3. Eigenvalue Problem

By Rayleigh’s principle (Golub & Van Loan, 1996) Equation 1 is minimized (for normalised coefficients

of Equation 2 satisfying  $\sum_j^n \pi_j^2 = 1$ ) by the eigenvector  $\pi^*$  with the most negative eigenvalue  $\lambda^*$  of the following non positive definite eigenvalue problem:

$$(K_{vx}K_{vx}^\top + \lambda_\Omega K_\Omega - \lambda_e K_e - \lambda_\nabla K_\nabla)\pi^* = \lambda^* \pi^*$$

which is in our case is typically a large system with a high sparsity ratio, which we solve using the JDQR algorithm (Sleijpen & Van der Vorst, 1996). Actually, we found that it was numerically more appropriate to add a multiple<sup>3</sup> of the identity in order to yield an equivalent positive definite system, and then solve for the smallest magnitude eigenvalue rather than the most negative.

As we have noted, by using a compactly supported kernel function  $\phi(\cdot)$ , all of the matrices that comprise the above eigenvalue problem will also be sparse. Moreover, they can be constructed efficiently using standard algorithms for range searching – a problem closely related to nearest neighbour searching. Range search algorithms typically use a spatially hierarchical tree like data structure in order to efficiently determine which points lie within some range of a query point.

Since we are minimising two terms and maximising another two, it may seem natural to minimise a ratio of sums of vector-matrix-vector products, as this leads to a positive definite generalised eigenvalue problem. Unfortunately this costs us an important degree of freedom in balancing the individual terms – minimising  $a/b$  is the same as minimising  $(\log(a) - \log(b))$ , so that if we balance the combination using a parameter  $c$  by minimising  $(\log(a) - c \log(b))$  we are equivalently minimising  $a/b^c$ . In other words, many Rayleigh’s quotient type problems probably ought to have an exponent in the denominator, but as this no longer corresponds to an equivalent eigenvalue problem, it will usually be considerably more difficult to solve. There are cases however in which both the numerator and denominator have equivalent units so that one may argue in favour of the ratio problem on the grounds that it measures a “unitless” ratio – popular among physicists, but here this is not the case.

## 4. Experiments

Implicit shape modelling usually applied to 3D data, however it is useful to experiment in 2D since we can then visualise the embedding function as a two-dimensional colour intensity plot. The following subsection investigates the behaviour of the system in two dimensions for various parameter settings.

<sup>3</sup>Of magnitude  $\lambda_e \|K_e\|_2 + \lambda_\nabla \|K_\nabla\|_2$ , where  $\|\cdot\|_2$  denotes the matrix spectral norm.

#### 4.1. Two Dimensional Toy Problem

Some specific examples of the qualitative nature of the solutions that arise are given in Figure 3, which demonstrates the nature of the energy/derivative balance. In general we found that the derivative term was necessary to avoid the type of behaviour shown on the bottom left of the figure. We also did some “grid searches” over the three parameters  $\lambda_e$ ,  $\lambda_\nabla$  and  $\lambda_\Omega$  in order to better understand how the optimisation problem behaves. Note that the problem of parameter choice is far more benign than in a classification problem, since we can make some reasonable measures of the quality of the solution without requiring a holdout set or cross-validation process. Two useful measures are defined as follows: given an embedding function  $f$  that models our training data set  $\mathcal{S}$ , we take an approximately uniform sampling  $\mathcal{R} \subseteq f^{-1}(0)$  of the zero set and compute the following:

$$m_{\mathcal{R}\mathcal{S}} = \max_{\mathbf{x}_1 \in \mathcal{R}} \min_{\mathbf{x}_2 \in \mathcal{S}} \|\mathbf{x}_1 - \mathbf{x}_2\|$$

$$m_{\mathcal{S}\mathcal{R}} = \max_{\mathbf{x}_1 \in \mathcal{S}} \min_{\mathbf{x}_2 \in \mathcal{R}} \|\mathbf{x}_1 - \mathbf{x}_2\|$$

These are, roughly speaking, the largest distance from  $\mathcal{R}$  to  $\mathcal{S}$  and *vice versa*. Both of these measures are useful, and if either measure is large then the solution cannot be a good one. It turns out that both the energy and the gradient term are necessary for maximum topological stability of the solution. As evidenced by Figure 4 however, too great a value of  $\lambda_\nabla$  leads to instability, while increasing  $\lambda_e$  reduces this effect, the net result being smoother but less precise solutions. We stress however that these measures are a heuristic and imperfect measure of the quality of solution – superior being a visual inspection. As expected, the regularisation parameter  $\lambda_\nabla$  controls the tradeoff between smoothness and fidelity to the data.

Note that Figure 3 also reveals how we chose our basis function centers – we use a coarse grid that covers the entire region of interest, and increasingly fine grids confined within some range of the data, using a multiple of the grid width as the basis function support.

#### 4.2. Three Dimensional Results

We also ran our system on some three dimensional data sets, particularly focussing on two important applications of the algorithm that for reasons discussed in Section 2 are difficult to achieve using the Slab SVM. The first is the filling or interpolation of holes in the data, as in Figure 5. This is a useful property as such holes often arise due to occlusions. Using a 3GHz Pentium III processor, we fit the bunny model in approximately forty minutes. The drawing took a similar amount of time, although this is due to the fact that

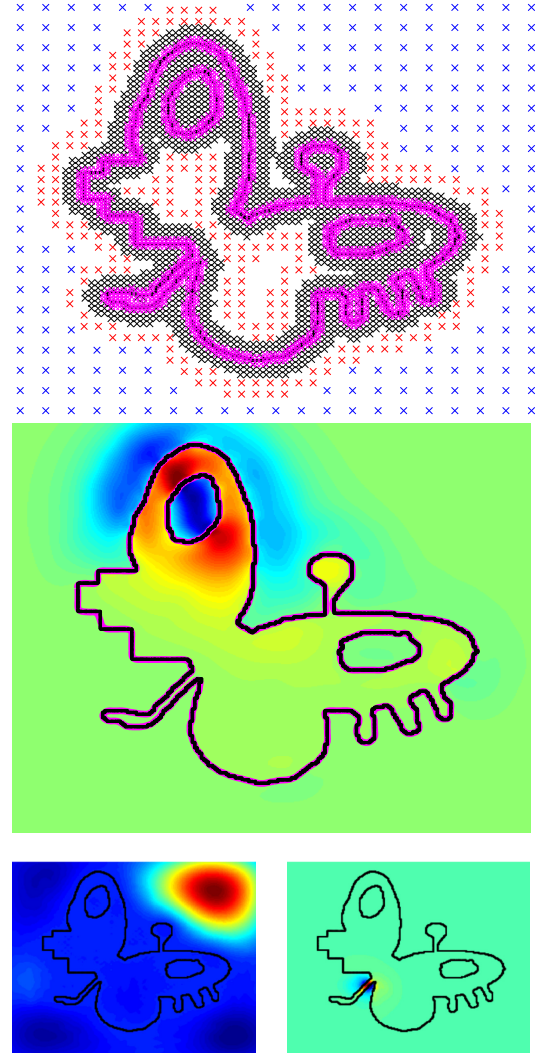


Figure 3. Investigating the role of the parameters in a two dimensional problem. *Top*: the data (black dots) and basis function centers (crosses), color-coded according to basis function width. The way we choose the basis is discussed in Section 4.1. There are a total of approximately 3K basis centers which we colored from widest to narrowest blue, red, black and magenta. *Middle*: a parameter set chosen using our automatic criteria (see Section 4.1) – visually there seems to be somewhat too much derivative term, but the zero level rendered as a magenta line is reasonable – as it is largely obscured by the data one must “zoom in” on an electronic copy of the document to see it properly. *Bottom-left*: too large an energy term results in a large smooth “bump” in the function outside the shape. *Bottom-right*: too large a gradient term results in a concentrated region of high gradient at some location on the data set, in this case in the bottom-left part of the shape.



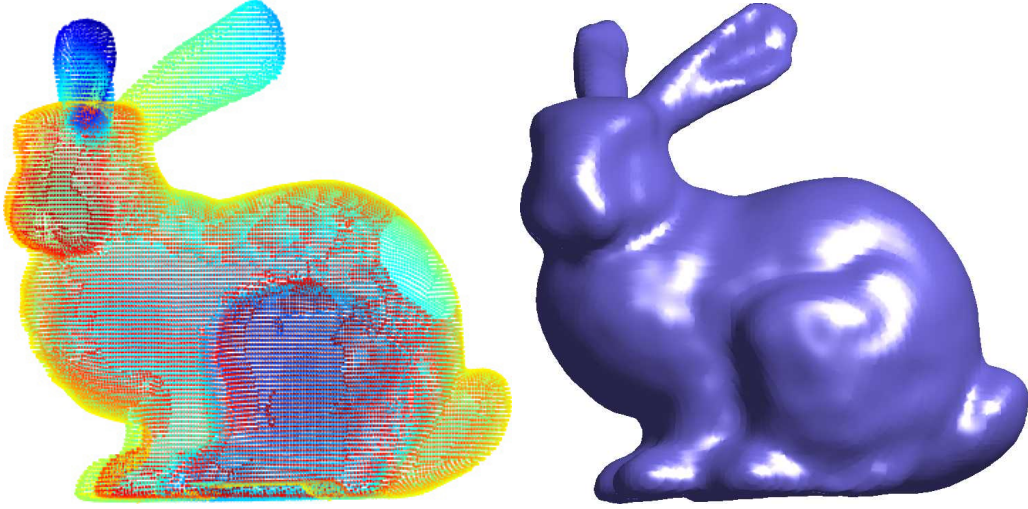


Figure 5. An example of smoothly interpolating missing data – we created a hole in the Stanford bunny data set (visible in the upper rear region of the bunny as shown on the left), resulting in a data set of 35K data points in three dimensions. The surface that we fit, using 14K basis functions, approximately recovers the original shape, as shown on the right.

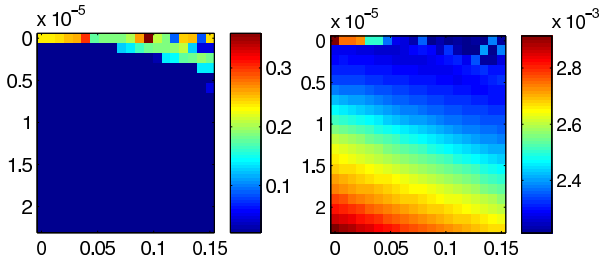


Figure 4. The heuristic evaluation measures  $m_{RS}$  (left) and  $m_{SR}$  (right) as defined in Section 4.1, over a range of values of  $\lambda_{\nabla}$  (on the horizontal axis) and  $\lambda_e$  (on the vertical axis), for constant  $\lambda_{\Omega}$  with a 2D data set.

we employed a marching cubes algorithm that naïvely evaluates the embedding function over the entire viewing region, rather than a faster method that follows the surface. We avoided such fast rendering schemes in order to be sure that there are no errant components of the zero set lying away from the data.

The second interesting property that we demonstrate is the ease with which set operations can be performed using implicit shape definitions. For example, given a pair of embedding functions  $f_1$  and  $f_2$  that are negative inside the shape and positive outside, the intersection of the shapes corresponds to the embedding function  $f_{\cap} = \max(f_1, f_2)$ . This is precisely how we generated Figure 6. Alternatively one can define an embedding function  $|f_1| + |f_2|$ , the zero level set of which is the intersection of the zero level sets of  $f_1$

and  $f_2$ . The fitting of the knot model took roughly fifteen minutes. The plain knot model was drawn in about forty minutes, whereas the intersection picture took approximately a day using a higher resolution.

To define the union of the two implicit shapes we can use  $f_{\cup} = \min(f_1, f_2)$ . Similarly to the intersection case, we can alternatively use the product  $f_1 f_2$ , the zero level set of which is the union of those of  $f_1$  and  $f_2$ , however this retains those components of the zero set interior to the resultant shape in spite of the fact that they will always be obscured from the observer.

## 5. Conclusions

We have proposed a method of fitting implicit shapes to a finite sampling of a manifold, without using normal vectors – an important problem since normals can be difficult to determine. The optimisation problem that we solve incorporates four intuitively justifiable terms, the balancing of which can be done automatically. We overcome the possible difficulties associated with the non-convexity of this problem by solving an equivalent eigenvalue problem which efficiently yields the globally optimal solution, and by using sparse basis functions of various widths the computational and storage requirements are reduced such that the method can handle many tens of thousands of points. It would also be interesting to try an objective function that is closer to ideal (such as forcing the gradient have approximately constant magnitude everywhere), in spite of the more difficult optimisation problem that ensues.

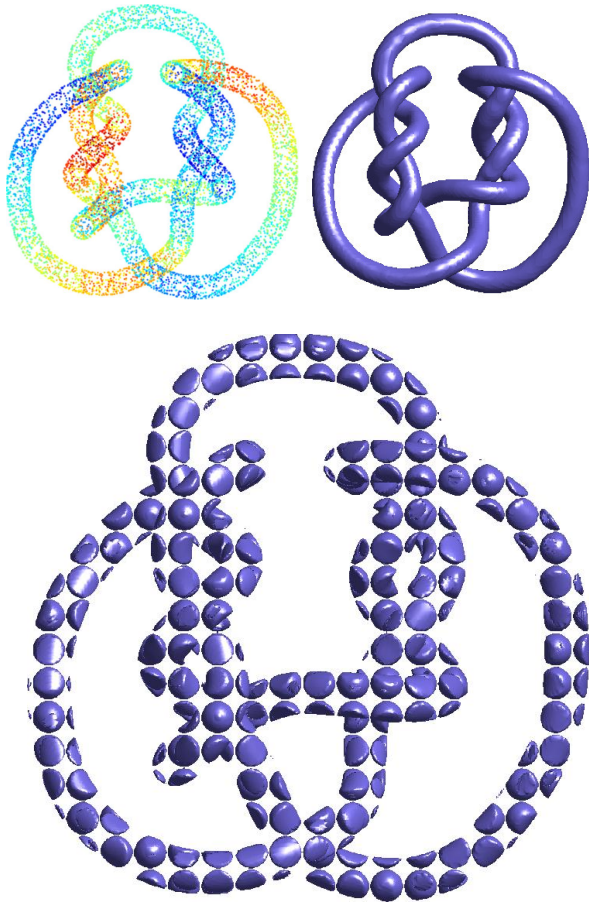


Figure 6. An example using the “knot” data set consisting of 10K data points in three dimensions, as shown on the top left (the colours have no special meaning). The implicit surface rendered on the top right was constructed using approximately 5K basis functions. The lower image demonstrates that set operations such as intersection are trivial with implicit shape definitions, regardless of the topological complexity that results – we intersected the knot shape with a grid of balls (also defined as an implicit).

## References

- Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., & Silva, C. T. (2001). Point set surfaces. *IEEE Visualization 2001* (pp. 21–28). San Diego, California: IEEE Computer Society.
- Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., & Evans, T. R. (2001). Reconstruction and representation of 3d objects with radial basis functions. *ACM SIGGRAPH 2001* (pp. 67–76). ACM Press.
- Duchon, J. (1976). Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *RAIRO Analyse Numrique*, 10, 5–12.
- Girosi, F., Jones, M., & Poggio, T. (1993). *Priors stabilizers and basis functions: From regularization to radial, tensor and additive splines* (Technical Report AI 1430). Massachusetts Institute of Technology.
- Golub, G. H., & Van Loan, C. F. (1996). *Matrix computations*. Baltimore MD: The Johns Hopkins University Press. 2nd edition.
- Ohtake, Y., Belyaev, A., & Seidel, H.-P. (2003). A multi-scale approach to 3d scattered data interpolation with compactly supported basis functions. *Proc. SMI 2003* (p. 292). IEEE Computer Society.
- Ohtake, Y., Belyaev, A., & Seidel, H.-P. (2004). Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph.*, 23, 609–612.
- Schölkopf, B., Giesen, J., & Spalinger, S. (2005). Kernel methods for implicit surface modeling. In L. K. Saul, Y. Weiss and L. Bottou (Eds.), *Advances in neural information processing systems 17*. Cambridge, MA: MIT Press.
- Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. Cambridge MA: MIT Press.
- Shen, C., O’Brien, J. F., & Shewchuk, J. R. (2004). Interpolating and approximating implicit surfaces from polygon soup. *ACM SIGGRAPH 2004*. Los Angeles, California: ACM Press.
- Sleijpen, G. L. G., & Van der Vorst, H. A. (1996). A Jacobi-Davidson iteration method for linear eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 17, 401–425.
- Smola, A., & Schölkopf, B. (1997). On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, 22, 211–231.
- Vapnik, V. (1998). *Statistical learning theory*. New York: John Wiley and Sons, inc.
- Walder, C., Lovell, B., & Kootsookos, P. (2003). Algebraic curve fitting support vector machines. In *Dicta 2003*, 693–702. Sydney.